

c)

DMA DEVICE

Patent Number: JP2000298640
Publication date: 2000-10-24
Inventor(s): TANAKA SATOSHI
Applicant(s): RICOH CO LTD
Requested Patent: ☐ JP2000298640
Application Number: JP19990104228 19990412
Priority Number(s):
IPC Classification: G06F13/28
EC Classification:
Equivalents:

Abstract

PROBLEM TO BE SOLVED: To facilitate to recognize what number descriptor the currently transferred block is set to, to optionally set a time interval between the 1st block transfer completion and the next block transfer start and further to facilitate to manage the generation, setting and storage of a descriptor in a program.

SOLUTION: This DMA(direct memory access) device is provided with one or more descriptors 610/620 which respectively include transfer source address information 612/622 showing transfer information blocks and transfer byte number information 612/623 and a transfer controlling means (DMAC (DMA controller)) which reads transfer information designated by the descriptors' information from memories 630/640. In such a case, the transfer controlling means generates continuous interrupt in timing shown by a timer variable for each transfer of one block based on the information of one descriptor. Then, it has 'unused' data for its management in each descriptor.

Data supplied from the esp@cenet database - I2

c)

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号
特開2000-298640
(P2000-298640A)

(43)公開日 平成12年10月24日(2000.10.24)

(51)Int.Cl.⁷
G 0 6 F 13/28

識別記号
3 1 0

F I
G 0 6 F 13/28

テーマコード(参考)
3 1 0 E 5 B 0 6 1

審査請求 未請求 請求項の数3 O L (全 6 頁)

(21)出願番号 特願平11-104228

(22)出願日 平成11年4月12日(1999.4.12)

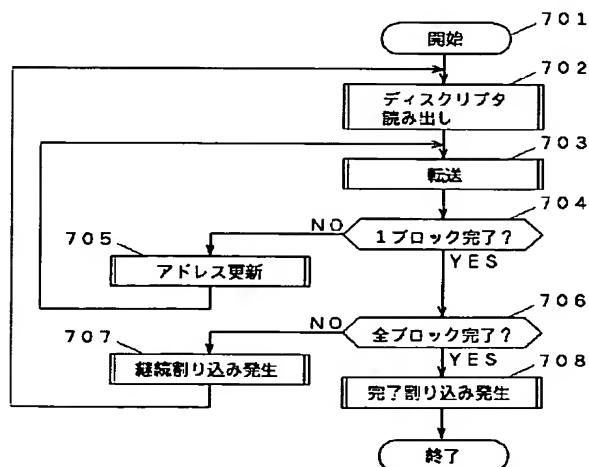
(71)出願人 000006747
株式会社リコー
東京都大田区中馬込1丁目3番6号
(72)発明者 田 中 智
東京都大田区中馬込1丁目3番6号 株式
会社リコー内
(74)代理人 100076967
弁理士 杉信 興
Fターム(参考) 5B061 BA02 BA03 DD01 DD06 DD11

(54)【発明の名称】 DMA装置

(57)【要約】

【課題】 現在、転送されているブロックが何番目のディスクリプタに設定されているブロックなのかの認識を容易にする。1つ目のブロック転送完了と次のブロック転送の開始までの時間を任意に設定可とする。ディスクリプタの生成、設定、保存をプログラムにおいて管理しやすくする。

【解決手段】 それぞれが、転送情報のブロックを示す、転送元アドレス情報612/622および転送バイト数情報613/623を含む、1以上のディスクリプタ610/620;および、これらのディスクリプタの情報が指定する転送情報をメモリ630/640から読出す転送制御手段400;を備えるDMA装置において、転送制御手段は、1つのディスクリプタの情報に基づいた1ブロックの転送毎に、タイマ変数が示すタイミングで、継続割り込みを発生する、ことを特徴とする。ディスクリプタ毎にその管理用の「未使用」データを有する。



【特許請求の範囲】

【請求項1】それぞれが、転送情報の転送単位区分を示す、一転送単位の転送元アドレス情報および転送バイト数情報を含む、1以上のディスクリプタ；および、これらのディスクリプタから順次に転送元アドレス情報および転送バイト数情報を読み出してこれらの情報に従って転送情報をメモリから読み出す転送制御手段；を備えるDMA装置において、

前記転送制御手段は、1つのディスクリプタの情報に基づいた一単位の転送毎に割り込み信号を発生する、ことを特徴とするDMA装置。

【請求項2】各ディスクリプタは割り込み発生タイミング情報を含み、転送制御手段は、1つのディスクリプタの情報に基づいた一単位の転送毎に該ディスクリプタが保持した割り込み発生タイミング情報が指定するタイミングで割り込み信号を発生する、請求項1記載のDMA装置。

【請求項3】各ディスクリプタは、それに宛てられた一単位の転送の制御には無関係な、ディスクリプタ管理用のデータを含む、請求項1又は請求項2記載のDMA装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ダイレクト・メモリ・アクセス（DMA）装置に関する。

【0002】

【従来の技術】特開平7-21117号公報には、DMAの1バスサイクル転送、2バスサイクル転送、バースト転送およびサイクルスチール転送が説明され、バースト転送には単転送モード、繰返し転送モード、アレイチェーン転送モードおよびリンクアレイ転送モードの4モードがあることが説明されている。

【0003】従来のディスクリプタ方式のコントローラ（DMAC）は、転送開始タイミングをカウンタのアンダーフローを使って制御することはできたが転送終了のタイミングを遅らせることはできなかった。また、ディスクリプタによって指定される転送対象のブロックを連続で転送することができても、ブロック転送完了で一旦停止し、再起動することはできなかった。

【0004】

【発明が解決しようとする課題】ディスクリプタを使った転送ではディスクリプタのチェーンが終了し、すべてのブロックが転送完了した場合にのみ、完了割り込みが発生していた。しかし、プリンタの描画処理のようにブロック毎の処理とブロック毎の転送を同時に行う場合、現在、転送されているブロックが何番目のディスクリプタに設定されているブロックなのかをプログラムが知ることはできなかった。周期的なブロック転送を異なるブロックに対して行うとき、転送先デバイスの処理が完了してから、次のブロックを送る必要がある。

【0005】本発明は、現在、転送されているブロック

が何番目のディスクリプタに設定されているブロックなのかを認識するための処理を容易にすることを第1の目的とし、1つ目のブロック転送完了と次のブロック転送の開始までの時間を設定できるようにすることを第2の目的とし、ディスクリプタの生成、設定、保存をプログラムにおいて管理しやすくすることを第3の目的とする。

【0006】

【課題を解決するための手段】（1）それぞれが、転送情報の転送単位区分（ブロック）を示す、一転送単位（1ブロック）の転送元アドレス情報（612/622）および転送バイト数情報（613/623）を含む、1以上のディスクリプタ（610/620）；および、これらのディスクリプタから順次に転送元アドレス情報および転送バイト数情報を読み出してこれらの情報に従って転送情報をメモリ（630/640）から読み出す転送制御手段（400）；を備えるDMA装置において、前記転送制御手段（400）は、1つのディスクリプタの情報に基づいた一単位の転送毎に割り込み信号（継続割り込み）を発生する、ことを特徴とするDMA装置。なお、理解を容易にするためにカッコ内には、図面に示し後述する実施例の対応要素の符号又は対応事項を、参考までに付記した。以下も同様である。

【0007】これによれば、ディスクリプタ毎に割り込み信号を発生するので、この信号に応答して、実行したディスクリプタの数をカウントアップして、何番目のディスクリプタに設定されているブロックの転送なのかを認識することが可能になる。また、割り込み信号に応答して次のブロックの転送を開始するか一時的に保留するかのチェックを行なって開始可のときに転送を開始するなど、順次転送を抑制することも可能になる。

【0008】

【発明の実施の形態】（2）各ディスクリプタは割り込み発生タイミング情報（モードワード614のタイマ変数）を含み、転送制御手段（400）は、1つのディスクリプタの情報に基づいた一単位の転送毎に該ディスクリプタが保持した割り込み発生タイミング情報が指定するタイミングで割り込み信号を発生する。

【0009】これによれば、割り込みのタイミングや継続のタイミングをディスクリプタ毎に設定することができる。

（3）各ディスクリプタは、それに宛てられた一単位の転送の制御には無関係な、ディスクリプタ管理用のデータ（モードワード614の「未使用」）を含む。

【0010】ディスクリプタ毎にディスクリプタ管理用のデータを用意することで、ソフトウェアによるディスクリプタの管理を容易に行なうことができる。

（4）各ディスクリプタが、割り込み信号の発生を指定するデータ（モードワード614のCONT）を含み、転送制御手段（400）は、各ディスクリプタから情報を読み込むときに該指定データをコントロールレジスタ（402）に書込んで、

情報を読込んだディスクリプタの情報に基づいた一単位の転送の終了時に、該指定データに基づいて割り込み信号を発生する。

(5) ディスクリプタ毎に割り込みを発生させることができるDMAコントローラ。

(6) ディスクリプタ毎に割り込み発生タイミングと再起動のタイミングを遅らせることができるDMAコントローラ。

(7) ディスクリプタ内に動作に無関係なデータ(モードワード614の「未使用」)を設けたことを特徴とするDMAコントローラ。

【0011】本発明の他の目的および特徴は、図面を参照した以下の実施例の説明より明らかになろう。

【0012】

【実施例】図1に本発明の一実施例の構成を示す。CPU100はマイクロプロセッサである。ROM200はCPU100が実行するプログラムを格納するリードオンリーメモリである。メモリコントローラ300はRAM600の制御信号を発生し、CPU100やDMAコントローラ400の要求に応じてRAM600に対してリードやライトを行う回路である。DMAコントローラ400が、本発明の一実施例のDMA装置の主要部である。I/Oデバイス500はたとえばハードディスクのようなデータ転送の対象となるデバイスである。RAM600はリードライト可能なランダムアクセスメモリである。

【0013】DMAを行う前に、CPU100がRAM600にディスクリプタを用意する。図2に、RAM600上に2つのディスクリプタ610、620を生成した状態を示し、図3の(a)に、1つのディスクリプタ610の構造を示す。図2に示すようにディスクリプタを2つ生成したとすると、第1のディスクリプタ610のネクストディスクリプタポインタレジスタ611(メモリの1領域:1つのレジスタ)には、第2のディスクリプタ620の先頭アドレスが格納されている。

【0014】第2のディスクリプタ620の転送元アドレスレジスタ612のデータは、バッファメモリ①630の先頭のアドレスを示す。バッファメモリ①630には転送すべきデータが格納されている。第1のディスクリプタ610の転送バイト数レジスタ613には、バッファメモリ①630に格納されたデータのうち、転送すべきバイト数が格納されている。

【0015】第1のディスクリプタ610のモードワードレジスタ614の、最下位ビットすなわちビットNo. 0には、この第1のディスクリプタ610で示される転送が終了したら、割り込みを発生するかどうかをきめるフラグデータCONTが存在し、これが「1」のとき、DMAコントローラ400が割り込み信号を発生する。同様に、第2のディスクリプタ620に示されるネクストディスクリプタポインタレジスタ621には、もし、

あれば第3のディスクリプタの先頭アドレスが格納されるが、図示例では、第2のディスクリプタ620は最後のディスクリプタなので、終了であることをDMAコントローラ400に教えるために、「0」が格納される。第2のディスクリプタ620の転送元アドレスレジスタ622はバッファメモリ②640の先頭アドレスを示し、バッファメモリ②640には転送すべきデータが格納されている。第2のディスクリプタ620の転送バイト数レジスタ623にはバッファメモリ②640に格納されたデータのうち、転送すべきバイト数が格納されている。第2のディスクリプタ620のモードワードレジスタ624には、何も設定されていない。

【0016】従来のディスクリプタ方式のDMAでは、最終ディスクリプタで示されるブロックを転送完了すると割り込みを発生していた。本実施例では、図3に示すように、最終ディスクリプタの前のディスクリプタ610内に割り込みイネーブルビット「CONT」を設けることで、ディスクリプタ単位で割り込みを発生させることができるようにした。ディスクリプタのサイズは何ワードでも、機能に問題はないがここでは4ワードとし、ディスクリプタの先頭アドレスからの増分アドレスをオフセットとして図3の(a)に示す。ネクストディスクリプタアドレスレジスタ611は、ディスクリプタ610、620をチェーン構造にしてリンクしていけるように、次に実行すべきディスクリプタ620のアドレスを格納する。転送元アドレスレジスタ612のデータは、このディスクリプタ610がバッファメモリ①630からI/Oデバイス500への転送を行うものとして、RAMメモリ600上に確保された転送データ(630)の先頭アドレスを示す。転送バイト数レジスタ613のデータは転送すべき長さ(バイト数)を示す。

【0017】モードワードレジスタ614は、ディスクリプタ単位での動作モードを設定するためのワードを格納し、詳細は図3の(b)に示す。モードワードレジスタ614は、この実施例の場合32bitの長さをもつ。その最下位ビットであるビットNo. 0のデータCONTは、「0」と「1」の2つの状態を取ることができ、このBitデータCONTは割り込みイネーブルビットとして機能する。割り込みイネーブルビットとは、このビットが「1」に設定されているディスクリプタの示す転送が完了したら、割り込みを発生することを許可することを示す。

【0018】図4の(a)に、DMAC(DMAコントローラ)400の構成を示す。ディスクリプタポインタレジスタ401は、RAMメモリ600上に格納されたディスクリプタ(610/620)の先頭アドレスを格納するためのレジスタである。コントロールレジスタ402は、CPU100からDMA動作開始の指示をうけるためのレジスタである。アドレスレジスタ403は、ディスクリプタが保持する転送元アドレスを格納するも

のであり、転送カウンタ409は実際の転送バイト数をカウントするカウンタである。転送バイト数レジスタ410はディスクリプタで示される転送バイト数を格納するためのレジスタである。

【0019】DMA転送制御回路404は、アドレスレジスタ403の転送元アドレスと転送カウンタ409のカウンタ値（転送済バイト数）の和で示されるアドレスのメモリをアクセスし、実際の転送を行う回路である。モードレジスタ406はディスクリプタのモードワードレジスタ614、624のデータを格納するためのレジスタである。ディスクリプタ制御回路405は、ディスクリプタ610、620の読み込みを行う回路である。割込みイネーブル信号ライン407を介して、モードレジスタ406の割込み指示ビットCONTのデータ（「1」又は「0」）がDMA転送制御回路404に与える。これが転送完了時に割り込みを発生させるかどうかを知らせる信号である。ディスクリプタで示される転送が完了したときに、DMA転送制御回路404（DMAC400）は、割込み信号ライン408を介してCPU100に対して割り込みを知らせる信号を与える。

【0020】CPU100が、RAM600上に転送データおよびディスクリプタを設定し、そしてDMAC400のディスクリプタポインタレジスタ401に第1ディスクリプタの先端アドレスを書込み、コントロールレジスタ402の、図4の(c)に示すビットEXECに、「1」を書くと、これに応答してDMAC400が転送動作を開始する。

【0021】図5に、DMAC400の転送動作フローを示す。開始701は、CPU100がDMAC400のコントロールレジスタ402のEXECビットに「1」を立てることで始まり、DMAC400はまずディスクリプタ読み出し702に進む。すなわちメモリ上のディスクリプタを読み出しに行く。具体的には、DMAC400は、ディスクリプタポインタレジスタ401のデータが示すアドレス（ディスクリプタ610）の内容を4ワード読み出して、最初のワード（ネクストディスクリプタアドレス）を図4の(a)に示すネクストディスクリプタポインタレジスタ411へ格納し、2番目のワード（転送元アドレス）はアドレスレジスタ403へ格納し、3番目のワード（転送バイト数）は転送バイト数レジスタ410へ格納し、4番目のワード（モードワード）はモードレジスタ406およびコントロールレジスタ402へ格納する。その動作波形を図6に示す。

【0022】図6を参照すると、CLK信号はシステム、の動作の基本となるクロック信号を示す。TRZ_START信号はメモリに対するアクセスが開始されたことを示す。ADDRESS信号はメモリのアドレスを示す。DATA信号はメモリが出力するデータ信号を示す。RDWR信号はメモリに対するアクセスが読み出しか書き込みかを示す。IADDR信号はDMAC400

の内部のレジスタのアドレスを示す。IRDWR信号はDMAC内部のアクセスが読み出しか書き込みかを示す。IDATA信号はDMAC内部のデータを示す。IRDATA信号はIADDR信号で指定されたDMACの内部レジスタの内容を示す信号である。

【0023】図6の①のCLK信号立ち上がりでは、TRZ_START信号がアサートされて、アクセスが開始されたことを示し、同時にADDRESS信号が有効になり、RDWR信号がハイとなり、読み出しであることを示している。②のクロック立ち上がりでは、そのADDRESS信号を受けて、メモリはデータをDATA信号に出力している。同時にDMACは内部の取り込むべきレジスタのアドレスをIADDR信号に出力し、IRDWR信号は内部レジスタに対する書き込みであることを示している。そして、メモリのデータ（DATA）を内部のデータ（IDATA）として出力している。③のクロック立ち上がりではDMACの内部のデータ（IDATA）からIADDR信号で示される内部レジスタに取り込まれ、レジスタの内容がIRDATA信号に保持されている。④のクロック立ち上がりはなにもないアイドルサイクルを示す。⑤のクロック立ち上がりは次のアクセスの開始を示す。この図では一回のメモリアクセスは4クロックで行われる。以上により第1ディスクリプタ610の各レジスタ611～614の各データが、DMAC400の各レジスタ411、403、410、406に書込まれる。

【0024】次にDMAC400は、各レジスタ411、403、410、406のデータが規定するDMA転送（703）を開始し、1つのディスクリプタ（第1ディスクリプタ610）で指定されたブロックの転送が完了したかどうかチェックする（704）。ブロックの転送が完了していないければ、次の転送のためにアドレスを更新する（705）。ブロックの転送が完了するまで、転送（703）を繰り返す。ブロックの転送が完了すると、DMAC400は、ディスクリプタそのもののチェーンが完了した、すなわち全体の転送が完了した、かどうかをチェックする（706）。

【0025】先に述べたが図示例では、第2のディスクリプタ620が最後のディスクリプタなので、終了であることをDMAコントローラ400に教えるために、「0」がネクストディスクリプタポインタ621に書込まれており、今転送を実行したディスクリプタが、仮に最後のディスクリプタ（第2ディスクリプタ620）であるとする、DMAC400のネクストディスクリプタポインタレジスタ411にはデータ「0」が書込まれているので、このデータに基づいて、全体の転送が完了した、と判定する（706）。

【0026】しかし、今終えた転送が第1ディスクリプタ610のデータに基づくものであると、ネクストディスクリプタポインタレジスタ411には第1ディスクリ

プタ610のネクストディスクリプタポインタ611のデータ(第2ディスクリプタ620の先頭アドレス)があるので、DMAC400は、次のディスクリプタ(620)が有効であるので、モードレジスタ406に格納している、元はモードワードレジスタ614にあったビットCONTの「1」をライン407経由で読んで、それが「1」であるので、コントロールレジスタ402に書込んでいるタイマ変数(元はモードワードレジスタ614にあったデータ)を計時タイマに設定して計時を開始し、タイマ変数が表わす時間が経過すると、継続の割込みを発生する(707)。すなわち、DMAC400は、コントロールレジスタ402のビットEXECを「0」にクリアして、CPU100への割込要求ライン408に割込み信号を与える(707)。そして、CPU100がこれに応答してコントロールレジスタ402のビットEXECを「1」にするのを待つ。そして、「1」になると、第2ディスクリプタ620のデータを読込んでレジスタ各レジスタ411、403、410、406、402に書込み(702)、転送(703)を行なう。

【0027】第2ディスクリプタ620のデータに基づいた転送を完了すると、ネクストディスクリプタポインタレジスタ411のデータが「0」になっているので、DMAC400は、完了割込み信号を発生して(708)、全処理を完了する。

【0028】継続割込み(707)は、DMAC400のコントロールレジスタのCONTビットが「1」(第1ディスクリプタ610)のとき、有効となるが「0」(第2ディスクリプタ620)のときは、継続割込みは発生しない。最終ディスクリプタ(620)中のモードワード(624)にもCONTビット相当のビットは存在するが、それは「0」とされ、コントロールレジスタ402のCONTビットが「0」になるので、DMAC400は、継続割込みは発生せず、完了割込み信号を発生する。

【0029】上述の実施例では、図3の(b)に示すように、ディスクリプタのモードワードレジスタ614内

にタイマ変数を用意して、1ブロックの転送完了後タイマ変数で指定した時間待ってから、継続割込みを発生する。このタイマ変数にて、1ブロック転送完了から継続割込み発生までの時間を設定することができる。最終ディスクリプタ(620)のモードワードレジスタ(624)にもタイマ変数を含めて、最終ブロックの転送完了からその時間分の遅延の後に完了割込み信号を発生するようにすることによって、完了割込みのタイミングも設定可能になる。図3の(b)に示すように、ディスクリプタのモードワードレジスタ614内に、ディスクリプタによる転送制御には無関係の「未使用」ビットを用意し、DMAC400がこれを使わないようにすることで、ソフトウェアによる該「未使用」ビットへの読み書きが可能であり、ソフトウェアによるディスクリプタの管理に使用することができる。

【図面の簡単な説明】

【図1】 本発明の一実施例のDMAシステムの構成を示すブロック図である。

【図2】 図1に示すRAM600上に設定されたディスクリプタを模式的に示す図表である。

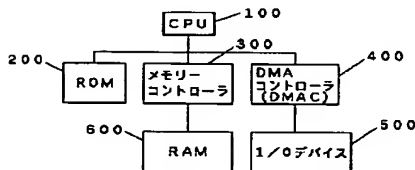
【図3】 (a)は図2に示す第1ディスクリプタ610のワード構成を模式的に示す図表、(b)は(a)に示すモードワードレジスタ614のデータ構成を示す図表である。

【図4】 (a)は図1に示すDMAコントローラ400の機能構成を示すブロック図、(b)は(a)に示すディスクリプタポインタレジスタ401の格納データの内容を示す図表であり、(c)はコントロールレジスタ402の格納データの内容の一部を示す図表である。

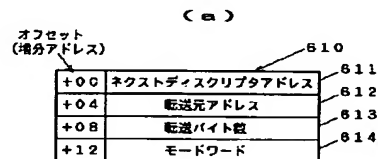
【図5】 図4に示すDMAコントローラ400のDMA転送制御の概要を示すフローチャートである。

【図6】 図4に示すDMAコントローラ400がRAM600からデータを読込むときの読み書きデータおよび制御信号の発生タイミングを示すタイムチャートである。

【図1】



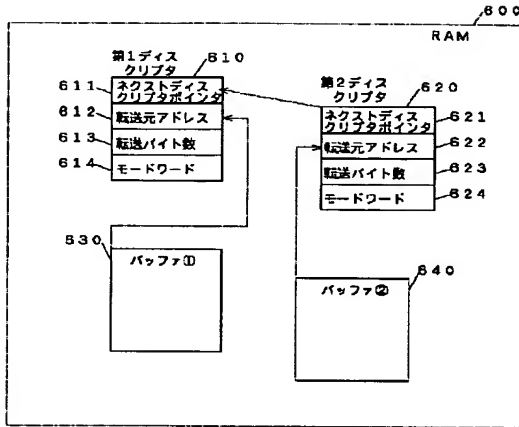
【図3】



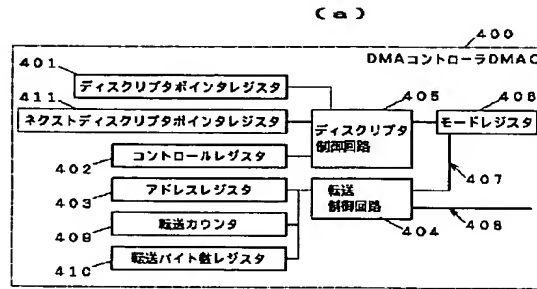
(b)

モードワード									
ビットNo.	31	...	16	17	...	2	1	0	
データの内容	タイマ変数			未使用				CONT	

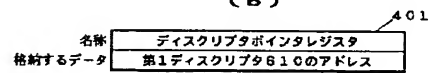
【図2】



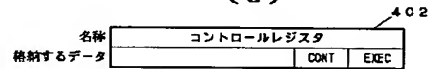
【図4】



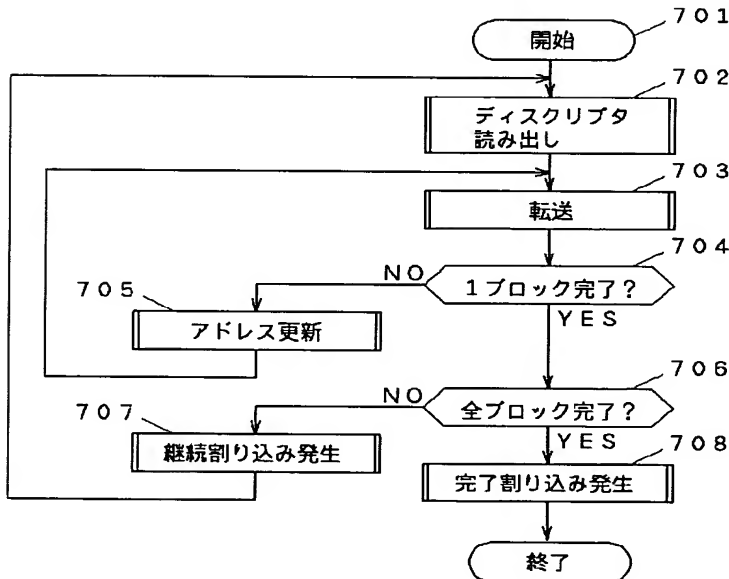
(b)



(c)



【図5】



【図6】

